Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 38(2), 2011, Pages 87–94
ISSN: 1223-6934, Online 2246-9958

Model-free Approaches in Learning the Multivariate Linear Regressive Models

Iuliana Paraschiv-Munteanu

Abstract. Learning from data means to have a learning method that is an algorithm implemented in software that estimates an unknown dependency between a system's inputs and outputs from the available data, namely from known samples. Once such a dependency has been accurately estimated, it can be used for prediction of future system outputs for known input values. The paper provides a series of results concerning the learning from data a linear regressive model in a multivariate framework. The parameter estimates of the regressive model are determined using the maximum likelihood principle and the adaptive learning algorithms are derived using the gradient ascend technique. In the second section of the paper the parameters of the linear regressive model are determined by minimizing the arithmetic mean of square errors and an adaptive learning scheme of gradient descent type is also considered. We consider a probabilistic approach in the third section for modeling the effects of both the latent variables and noise. The cumulative effects of latent variables and noise are modeled in terms of multivariate Gaussian distributions. The predicted output is expressed as the sum of a linear combination of the entries of the input and the random vector that represents the effects of the unobservable factors and noise. The parameters of the regressive model are estimated by maximizing the likelihood function for given finite length sequence of observations, and an adaptive learning algorithm of gradient ascent type is proposed in the final part of the section. A series of concluding remarks and suggestions for further work are formulated in the final section of the paper.

2010 Mathematics Subject Classification. Primary 62J05; Secondary 62J12.
Key words and phrases. linear regressive models, learning from data, supervised learning, maximum likelihood principle, adaptive learning.

1. Introduction

Machine Learning deals with programming computers to optimize a performance criterion on the basis of finite sets of example data or past experience. Dealing with the design of algorithms and techniques that allow machines to learn in the sense that they improved the performance through experience, machine learning can be viewed as a branch of artificial intelligence ([1], [6]).

The tremendous growth in practical applications of machine learning over the past decade has been accompanied by a wide variety of important developments in the underlying algorithms and techniques that make use of concepts and results coming from several areas as mathematical statistics, computer science and engineering.

We consider the learning environment described in Figure 1 ([2]). Let **S** be a system that for any n-dimensional input x computes an m-dimensional output y according to an unknown law. In the simplest approach we can assume that the output y is uniquely determined by the input x. However, the output can be influenced by a series of

Received February 20, 2011. Revision received May 11, 2011.
The presented work was developed in the framework of the Ph.D. program al University of Pitesti, Romania.

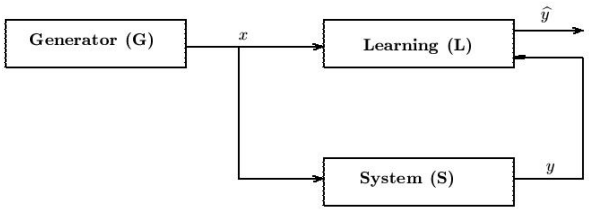

FIGURE 1. The scheme of learning environment.

unobservable factors, and the dependency between the inputs and outputs of **S** could be of non-deterministic type. Consequently, in a more sophisticated approach we are forced to take into account a non-deterministic dependency, modelled for instance in probabilistic terms, as a reasonable hypothesis concerning the unknown law. In our model we consider the Generator, denoted by **G**, the source that generates the inputs. Mainly, there are two ways to model **G**, namely when the mechanism of generating inputs is known by the observer and when the law according to which the inputs are generated is also unknown, respectively. The third component of our learning environment denoted by **L**, is responsible with possible models of the unknown dependency corresponding to **S**. The learning component **L** implements a class of hypothesis (models) Ω, such that to each particular hypothesis $\omega \in \Omega$ corresponds a function $\varphi_\omega : \mathcal{X} \to \mathcal{Y}$ defined on the space of inputs $\mathcal{X}$ and taking values in the space of outputs $\mathcal{Y}$. For each particular input x_0, $\widehat{y}_0 = \varphi_\omega(x_0)$ is the estimate of the **S**'s output corresponding to x_0 in case of the model ω. Being given a criterion function $\mathcal{C}$ that expresses numerically the fitness of each model with respect to the available evidence E, about **S**, the best model $\omega_0(E)$ is a solution of the optimization problem

$$\arg\left(\text{optimize}_{\omega\in\Omega}\, \mathcal{C}(\omega, E)\right) . \tag{1}$$

In the case of supervised learning the available evidence E is represented by a finite set of pairs $\{(x_i, y_i)\ , 1 \le i \le N\} \subset \mathcal{X} \times \mathcal{Y}$, where each y_i is the actual output of **S** for the input x_i. If we assume that the unknown dependency is of deterministic type, that is the inputs and the outputs of **S** are functionally related a reasonable choice of the criterion function $\mathcal{C}$ is the arithmetic mean of the square errors, that is for each $\omega \in \Omega$,

$$\mathcal{C}(\omega, E) = \frac{1}{N}\sum_{i=1}^{N} \|y_i - \varphi_\omega(x_i)\|^2 . \tag{2}$$

The optimization problem (1) becomes

$$\arg\left(\min_{\omega\in\Omega} \mathcal{C}(\omega, E)\right) , \tag{3}$$

and its solutions are called the Minimum Square Errors (MSE) models computed on the basis of $\{(x_i, y_i)\ , 1 \le i \le N\}$.

In case we adopt a more complex approach by including the effects of possible existing latent variables, each hypothesis $\omega \in \Omega$ corresponds to a probabilistic model for the latent vector. For simplicity sake, we consider that the latent vector is a continuous random vector, that is to each $\omega \in \Omega$ corresponds a conditional density function $f(\cdot|\cdot, ; \omega)$. Put in other words, for each $\omega \in \Omega$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $f(y|x\,; \omega)$ expresses 'the chance' of getting the output y for the input x in case of the model ω. If the available evidence about **S** is $\{(x_i, y_i)\ , 1 \le i \le N\}$ then a reasonable choice

of $\mathcal{C}(\omega, E)$ is the likelihood function. If we assume that the inputs $x_1, \ldots, x_N$ are independently generated by $\mathbf{G}$ then

$$\mathcal{C}(\omega, E) = \prod_{i=1}^{N} f\left(y_i | x_i \,;\, \omega\right) , \tag{4}$$

and the optimization problem (1) becomes

$$\arg\left(\max_{\omega \in \Omega} \mathcal{C}(\omega, E)\right) . \tag{5}$$

The solutions of (5) are the Maximum Likelihood (ML) models computed on the basis of $\{(x_i, y_i)\ , 1 \leq i \leq N\}$.

2. Modeling the Learning component in terms of linear hypothesis

Let $\mathcal{P}$ be the procedure used by the learning component, $\mathbf{L}$, to extract information from the available data $(x_1, y_1), \ldots, (x_N, y_N)$ in order to compute an approximation of the actual but unknown dependency between the inputs and the outputs of $\mathbf{S}$. If we denote by $\widehat{\omega}$ the model computed by $\mathcal{P}$ then $\widehat{y} = \varphi_{\widehat{\omega}}(x)$ is the predicted output if the input x is applied to $\mathbf{S}$ where, $\varphi_{\widehat{\omega}} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. The simplest class of hypothesis is the linear class where each individual hypothesis corresponds to a linear transform. In this case for each model $\omega \in \Omega$, the predicted output is $\varphi_\omega(x) = \beta^T \begin{pmatrix} 1 \\ x \end{pmatrix}$, $\beta \in \mathcal{M}_{(n+1)\times m}(\mathbb{R})$. From the point of view of the MSE (Minimum Square Errors) criterion, being given the data $(x_1, y_1), \ldots, (x_N, y_N)$, the optimal model is $\widehat{\omega} = \omega_{\widehat{\beta}_{MSE}}$ where

$$\widehat{\beta}_{MSE} = arg\left(\inf_{\beta \in \mathcal{M}_{(n+1)\times m}(\mathbb{R})} F_N(\beta)\right) , \tag{6}$$

and $F_N(\beta) = \dfrac{1}{N}\sum_{i=1}^{N} \left\| y_i - \beta^T z_i \right\|^2$, $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $1 \leq i \leq N$.

Using straightforward computations we get

$$F_N(\beta) = \operatorname{tr}\left(\widehat{P}_N\right) - 2\operatorname{tr}\left(\beta^T \widehat{Q}_N\right) + \operatorname{tr}\left(\beta^T \widehat{S}_N \beta\right) ,$$

where

$$\widehat{P}_N = \frac{1}{N}\sum_{i=1}^{N} y_i y_i^T \in \mathcal{M}_m(\mathbb{R}) , \quad \widehat{Q}_N = \frac{1}{N}\sum_{i=1}^{N} z_i y_i^T \in \mathcal{M}_{(n+1)\times m}(\mathbb{R})$$

$$\text{and } \widehat{S}_N = \frac{1}{N}\sum_{i=1}^{N} z_i z_i^T \in \mathcal{M}_{n+1}(\mathbb{R}).$$

If we denote by $Z = (z_1, \ldots, z_N)$ and $Y = (y_1, \ldots, y_N)$ the matrices of augmented inputs and corresponding outputs respectively, we get the compact forms,

$$NF_N(\beta) = \operatorname{tr}\left(\left(Y - \beta^T Z\right)\left(Y - \beta^T Z\right)^T\right),\ N\widehat{P}_N = YY^T,\ N\widehat{Q}_N = ZY^T,\ N\widehat{S}_N = ZZ^T.$$

Theorem 2.1. *([7], [8]) The MSE estimation for parameter β for given data (x_1, y_1), $(x_2, y_2), \ldots, (x_N, y_N)$, is*

$$\widehat{\beta}_{MSE} = \left(Y Z^+\right)^T . \tag{7}$$

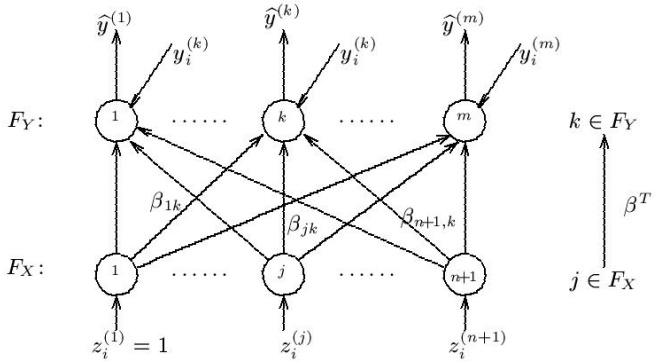

FIGURE 2. Architecture feed-forward type neural model of the System.

Obviously, $\widehat{S}_N^+ = N\left(Z^T\right)^+ Z^+$ and $\widehat{S}_N^+\widehat{Q}_N = \left(YZ^+\right)^T$, therefore $\widehat{\beta}_{MSE} = \left(YZ^+\right)^T = \widehat{S}_N^+\widehat{Q}_N \in \mathcal{S}$. For a new input x, the best prediction about the output of $\mathbf{S}$ computed on the basis of the available data is $\widehat{y} = \widehat{\beta}_{MSE}x$, that is

$$\widehat{\beta}_{MSE} = \widehat{S}_N^+\widehat{Q}_N = \left(YZ^+\right)^T . \tag{8}$$

is the "best model" for $\mathbf{S}$ computed on the basis of the available data.

Adaptive learning based on data $(x_1, y_1), \ldots, (x_N, y_N)$ can be done using gradient descent methods and/or stochastic gradient methods.

A learning scheme obtained using the gradient descent method ("batch") updates the parameter β according to the rule

$$\beta^{new} \leftarrow \beta^{old} + \rho\left(Q - S\beta^{old}\right).$$

The stochastic gradient method is the sequential version of the gradient descent procedure. The advantages of using the stochastic gradient method scheme, on one hand from its computational simplicity, and on the other hand from the locality feature that allows implementation on a simple feed-forward neural network ([4]). The stochastic gradient descent learning scheme is briefly,

Input: $(x_1, y_1), \ldots, (x_N, y_N)$
Initializations: β_0, $\rho > 0$, $\mathcal{C}$, $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $1 \le i \le N$;
$\beta^{old} \leftarrow \beta_0$
repeat
  $\beta = \beta^{old}$
  for $i \leftarrow 1, N$
    for $k \leftarrow 1, m$
      for $j \leftarrow 1, n+1$

$$\beta_{jk}^{new} \leftarrow \beta_{jk} + \rho z_i^{(j)}\left(y_i^{(k)} - \sum_{p=1}^{n+1} z_i^{(p)}\beta_{pk}\right)$$

      end for
    end for
    $\beta \leftarrow \beta^{new}$
  end for
  evaluate $\mathcal{C}$
  $\beta^{old} \leftarrow \beta^{new}$
until $\mathcal{C}$
Output: β^{new}

where $\mathcal{C}$ is a stopping condition and a learning rate ρ is a conventionally selected positive number.

The stochastic gradient learning algorithm can be implemented on a simple two-layer feed-forward neural architecture where the input layer F_X and the output layer F_Y consist of $n+1$ and m neurons respectively, the synaptic memories of the neurons in the output layer F_Y being the columns of the currently computed matrix β. The scheme is presented in Figure 2.

During the learning process, if (x_i, y_i) is the current test example, $x_i = \left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\right)$, $y_i = \left(y_i^{(1)}, \ldots, y_i^{(m)}\right)$, then the entries of $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$ are applied as inputs to the neurons of F_X and the entries of y_i are combined at the level of each neuron k of F_Y to update its synaptic memory according to the rule

$$\beta_{jk} \leftarrow \beta_{jk} + \rho z_i^{(j)} \left(y_i^{(k)} - \sum_{p=1}^{n+1} z_i^{(p)} \beta_{pk} \right), \quad j = 1, \ldots, n+1 \,.$$

3. A Probabilistic Model for the System input-output dependency

An alternative approach to the modeling of System input-output dependency can be considered by postulating certain parametric expression for the conditional repartition of the outputs on the inputs, $f(y|x,\theta)$, denoting by Θ the domain of the parameter θ, $f(\cdot|\theta)$ is a possible model of the unknown dependency for each $\theta \in \Theta$, such that for each input x and output y, $f(y|x,\theta)$ is the probability that being given the input x and the model θ to obtain the output y. Several intuitively justified criteria can be considered in order to identify the "fittest" model on the basis of the available data $\{(x_i, y_i)\ , 1 \leq i \leq N\}$. In the following we consider the likelihood function $L(\theta, x_1, \ldots, x_N, y_1, \ldots, y_N)$ to express the quality of each model θ to explain the available data. Assuming that the data are independent,

$$L(\theta, x_1, \ldots, x_N, y_1, \ldots, y_N) = \prod_{i=1}^{N} f(y_i|x_i, \theta) \,,$$

an optimal model $\widehat{\theta}_{MLE}$ according to the principle of maximum likelihood being a solution of the optimization problem

$$\arg \left(\max_{\theta \in \Theta} \left(L(\theta, x_1, \ldots, x_N, y_1, \ldots, y_N) \right) \right) .$$

Usually, the outputs of the System are conditioned not only on the observable input, but by some unobservable latent variables as well as by noise. In other words, for each input x_i the output y_i of the System depends on x_i and on unobservable variables ε. If we consider a parametric expression to model the effect of x_i on y_i then $y_i = g(\beta, x_i) + \varepsilon$.

A natural extension of the model proposed in section 2 is to combine a linear dependency on the input entries to a Gaussian multi-variational model for the effect of noise and/or latent variables, that is the estimate of the unknown conditional repartition of the output of $\mathbf{S}$ on the input is $\widehat{p}(y|x) = \beta^T \begin{pmatrix} 1 \\ x \end{pmatrix} + h(\varepsilon)$, where h is the density function of the m-dimensional Gaussian repartition $h(\varepsilon) \sim \mathcal{N}(\mu, \Sigma)$. Put in other words, the estimate of the output of $\mathbf{L}$ for the input x_i is $\widetilde{y}_i = \beta^T z_i + \varepsilon$, where

$z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $\varepsilon \sim \mathcal{N}(\mu, \Sigma)$, $\mu \in \mathbb{R}^m$ and $\Sigma \in \mathcal{M}_m(\mathbb{R})$ is a symmetric positive definite matrix.

In this case, a particular hypothesis $\omega \in \Omega$ corresponds to a tuple (β, μ, Σ), the conditional repartition of the output of $\mathbf{L}$ being given the input x_i is modeled by the density function of $\widetilde{y}_i$. Obviously, in case of the normal model for the effect of noise and latent variables we get $\widetilde{y}_i \sim \mathcal{N}\left(\beta^T z_i + \mu, \Sigma\right)$, that is the expression of the density function in the hypothesis $\omega = (\beta, \mu, \Sigma)$ is,

$$f(y_i|x_i, \beta, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} \exp\left\{-\frac{1}{2}\left(y_i - \beta^T z_i - \mu\right)^T \Sigma^{-1}\left(y_i - \beta^T z_i - \mu\right)\right\}.$$

In the following will try to fit the best model using the paradigm of the maximum likelihood being given the set of examples $\{z_1, \ldots, z_N\}$. For simplicity sake, we consider that the latent variables are uncorrelated and of unit variance that is for all hypothesis $\Sigma = \mathbf{I}_m$ therefore $\omega = (\beta, \mu)$. The log-logarithm of the maximum likelihood function is

$$l(\beta, \mu) = l(\beta, \mu, x_1, \ldots, x_N, y_1, \ldots, y_N) =$$
$$-\frac{mN}{2}\ln(2\pi) - \frac{1}{2}\sum_{i=1}^{N}\left(\left(y_i - \mu\right)^T - z_i^T\beta\right)\left(\left(y_i - \mu\right) - \beta^T z_i\right),$$

therefore the best model is

$$\left(\widehat{\beta}_{MLE}, \widehat{\mu}_{MLE}\right) = \arg\left(\max_{\beta \in \mathcal{M}_{(n+1)\times m}(\mathbb{R}), \mu \in \mathbb{R}^m} l(\beta, \mu)\right) =$$
$$\arg\left(\min_{\beta \in \mathcal{M}_{(n+1)\times m}(\mathbb{R}), \mu \in \mathbb{R}^m} \sum_{i=1}^{N}\left(\left(y_i - \mu\right)^T - z_i^T\beta\right)\left(\left(y_i - \mu\right) - \beta^T z_i\right)\right)$$

Theorem 3.1. *The maximum likelihood estimates of the parameters β, μ are*

$$\widehat{\beta}_{MLE} = \left(Y(ZA)^+\right)^T, \quad \widehat{\mu}_{MLE} = \frac{1}{N}\left(Yu - Y(ZA)^+ Zu\right), \tag{9}$$

where $Y = (y_1, \ldots, y_N) \in \mathcal{M}_{m\times N}(\mathbb{R})$, $Z = (z_1, \ldots, z_N) \in \mathcal{M}_{(n+1)\times N}(\mathbb{R})$,
$u = (1, \ldots, 1)^T \in \mathbb{R}^N$, $A = \mathbf{I}_N - \frac{1}{N}uu^T$.

Proof. The entries of the gradient of the log-logarithm of the likelihood function with respect to (β, μ) are

$$\nabla_\beta l(\beta, \mu, x_1, \ldots, x_N, y_1, \ldots, y_N) = ZY^T - (Zu)\mu^T - ZZ^T\beta,$$
$$\nabla_\mu l(\beta, \mu, x_1, \ldots, x_N, y_1, \ldots, y_N) = Yu - N\mu - \beta^T(Zu),$$

that is the space of critical points is the set of the solution of the system

$$\begin{cases} ZY^T - (Zu)\mu^T - ZZ^T\beta = \mathbf{O}_{(n+1)\times m} \\ Yu - N\mu - \beta^T(Zu) = \mathbf{0}_m. \end{cases}$$

Since from the second vectorial equation we get

$$\mu = \frac{1}{N}\left(Yu - \beta^T Zu\right),$$

by replacing it in the first vectorial equation we obtain

$$\beta_0 = \left(ZAZ^T\right)^+\left(ZAY^T\right).$$

Using the obvious properties

$$A^2 = A = A^T \quad \text{and} \quad A^+ = A,$$

we obtain

$$\beta_0 = \left(Y\,(ZA)^{+}\right)^T$$

therefore, by replacing it in the expression of μ we get

$$\mu_0 = \frac{1}{N} Y\left(\mathbf{I}_N - Y\,(ZA)^{+} Z\right) u.$$

Using straightforward computations we obtain

$$l\,(\beta_0, \mu_0) = -\frac{mN}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(Y\left(A - (ZA)^{+} ZA\right)\left(A - (ZA)^{+} ZA\right)^T Y^T\right).$$

In order to prove that (β_0, μ_0) is the maxima point of the log-likelihood function, for arbitrary (β, μ) we get

$$l\,(\beta, \mu) = -\frac{mN}{2}\ln(2\pi) - \frac{1}{2}\mathrm{tr}\left(\left(Y - \mu u^T - \beta^T Z\right)\left(Y - \mu u^T - \beta^T Z\right)^T\right).$$

In order to compare the values of the log-likelihood function for (β, μ) and (β_0, μ_0) using the relations

$$\left(A - (ZA)^{+} ZA\right)\left(A - (ZA)^{+} ZA\right)^T = A - A\,(ZA)^{+}\,(ZA)\,,$$
$$Y - \mu u^T - \beta^T Z = Y\left(A - (ZA)^{+}\,(ZA)\right) + (\mu_0 - \mu)\,u^T + (\beta_0 - \beta)^T Z$$

we obtain

$$l\,(\beta, \mu) = l\,(\beta_0, \mu_0) - \frac{1}{2}\,\mathrm{tr}\left((\beta_0 - \beta)^T ZZ^T\,(\beta_0 - \beta)\right) -$$
$$\mathrm{tr}\left(Y\left(\mathbf{I}_N - (ZA)^{+} Z\right) Au\,(\mu_0 - \mu)^T\right) - \mathrm{tr}\left(Y\left(\mathbf{I}_N - (ZA)^{+} Z\right) AZ^T\,(\beta_0 - \beta)^T\right).$$

Obviously, $Au = \mathbf{0}_N$ and $\left(\mathbf{I}_N - (ZA)^{+} Z\right) AZ^T = \mathbf{O}_{N,n+1}$, therefore

$$l\,(\beta, \mu) = l\,(\beta_0, \mu_0) - \frac{1}{2}\,\mathrm{tr}\left((\beta_0 - \beta)^T ZZ^T\,(\beta_0 - \beta)\right) \leq l\,(\beta_0, \mu_0)\,. \qquad \square$$

The adaptive learning of the parameters μ and β may be installed using for instance the gradient ascent method yielding to the following learning algorithm

Input: $(x_1, y_1), \ldots, (x_N, y_N)$
Initializations: β_0, μ_0, $\mathcal{C}$, $\rho > 0$
Compute $z_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix}$, $1 \leq i \leq N$; $Q = \sum_{i=1}^{N} z_i y_i^T$; $S = \sum_{i=1}^{N} z_i z_i^T$
$\beta^{old} \leftarrow \beta_0\,, \quad \mu^{old} \leftarrow \mu_0$
repeat
$$\beta^{new} \leftarrow \beta^{old} + \rho\left(Q - \left(\sum_{i=1}^{N} z_i\right)\left(\mu^{old}\right)^T - S\beta^{old}\right)$$
$$\mu^{new} \leftarrow \mu^{old} + \rho\left(\left(\sum_{i=1}^{N} y_i\right) - \sum_{i=1}^{N}\left(\beta^{old}\right)^T z_i - N\mu^{old}\right)$$
evaluate $\mathcal{C}$
$\beta^{old} \leftarrow \beta^{new}$; $\mu^{old} \leftarrow \mu^{new}$
until $\mathcal{C}$
Output: β^{new}, μ^{new}.

4. Conclusion

In this paper are presented two linear regressive models: one is a linear regressive model determined by minimizing the arithmetic mean of square errors, and the other is a probabilistic model which includes effects of the latent variables and the noise. For both models we found exact solutions for the parameters. The novelty of this study is demonstration of the Theorem 3.1 which gives the exact expressions of the parameters for proposed probabilistic model. In the future we will try to generalize the probabilistic model by increasing the number of parameters.

References

[1] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, Massachusetts 2010.
[2] V. Cherkassky and F. Mulier, *Learning from Data Concepts, Theory, and Methods (Second Edition)*, John Wiley & Sons, Inc., 2007.
[3] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning - Data mining, Inference and Prediction (Second Edition)*, Springer-Verlag, 2009.
[4] S. Haykin, *Neural Networks:A Comprehensive Foundation*, Prentice Hall, 2000.
[5] A.K. Jain, R. Duin and J. Mao, Statistical Pattern Recognition: A Review, *IEEE Transactions On Pattern Analysis and Machine Intelligence* **22** (2000), No. 1, 4–37.
[6] S. Marsland, *Machine Learning: An Algorithmic Perspective*, CRC Press, Taylor & Francis Group, Boca Raton - London - New York, 2009.
[7] I. Paraschiv-Munteanu, Theoretical approach in performance evaluation of a classsification system, *University of Pitesti - Scientific Bulletin, Serie Mathematics and Computer Science* **14** (2008), 139–150.
[8] L. State, I. Paraschiv-Munteanu, *Introducere in teoria statistica a recunoasterii formelor*, Editura Universitatii din Pitesti, Romania, 2009.

(Iuliana Paraschiv-Munteanu) Faculty of Mathematics and Computer Science, University of Bucharest, 14 Academiei Street,Bucharest, 010014, Romania
E-mail address: pmiulia@fmi.unibuc.ro